Magistère de mathématiques

TP7 – Équations différentielles : résolution numérique

Dans tout ce TP, on n'utilisera jamais la bibliothèque `scipy.integrate` avec la fonction `odeint`. Le but du TP est de programmer des résolutions numériques d'équations différentielles « à la main ».

Exercice 1. *Schéma d'Euler explicite*

On considère un problème de Cauchy

$$\begin{cases} y' = f(t,y), \\ y(t_0) = y^0. \end{cases} \tag{1}$$

où f est une fonction définie sur un ouvert U de $\mathbb{R} \times \mathbb{R}^d$, à valeurs dans $\mathbb{R}^d$, $(t_0, y^0) \in U$ est la condition initiale. Sous l'hypothèse que f est continue sur U et *localement lipschitzienne* par rapport à sa seconde variable y, le **théorème de Cauchy-Lipschitz** garantit l'existence d'une unique solution maximale y, définie et continuement différentiable sur un intervalle I ouvert contenant t_0.

Pour simplifier, on se limitera au cas des équations autonomes (la fonction f ne dépendant alors pas de t) vérifiant de plus que $f \in \mathcal{C}^1(\mathbb{R}^d; \mathbb{R}^d)$ est de différentielle bornée, si bien que toutes les solutions sont globales et définies de manière unique par leur valeur en $t_0 = 0$.

En intégrant l'équation différentielle (1) entre deux instants t et $t+h$ où $h > 0$, on obtient la formulation

$$y(t+h) = y(t) + \int_t^{t+h} f(s, y(s))\mathrm{d}s. \tag{2}$$

Les méthodes numériques permettant de résoudre le problème (1) sont obtenues sur la base d'une formule d'approximation de cette intégrale par une méthode de quadrature.

La méthode d'**Euler explicite** définit une suite d'instants (t_i) et des approximations $y_i \in \mathbb{R}^d$ de la solution exacte $y(t_i)$. Elle se base sur une approximation de l'intégrale précédente par l'approximation des rectangles à gauche. La relation de récurrence prend alors la forme :

$$\begin{aligned} & y_0 = y^0, \\ & \forall i \in \mathbb{N}, \quad y_{i+1} = y_i + h_i f(t_i, y_i), \\ & \forall i \in \mathbb{N}, \quad t_{i+1} = t_i + h_i. \end{aligned} \tag{3}$$

Typiquement, considérant un intervalle $[0, t_f]$, on fixe un entier N. Le pas h_i est alors choisi égal à $h = t_f/N$ pour tout i, $0 \le i \le N$, de sorte que l'approximation se fait aux points de la subdivision uniforme $0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = t_f$.

1. Programmer cette méthode sur l'exemple élémentaire : $y' = -10y$ avec la donnée de Cauchy $(t_0, y^0) = (0, 1)$ sur l'intervalle de temps $[0, 5]$. Comparer graphiquement la solution approchée avec la solution exacte y, ceci pour différentes valeurs de l'entier N.
2. Vérifier expérimentalement que la convergence est au moins d'ordre 1 au sens où : il existe une constante C, dépendant de y^0, f et T seulement, telle que si on note $h = \sup_i(t_{i+1} - t_i)$, on a
$$\sup_{0 \le i \le N} |y(t_i) - y_i| \le Ch.$$

Application : équation de la chaleur instationnaire

On cherche à approcher la solution instationnaire de l'EDP

$$\frac{\partial u(t,x)}{\partial t} = \frac{\partial^2 u(t,x)}{\partial x^2} + s(x), \tag{4}$$

avec les conditions au bord $u(t,0) = u(t,1) = 0$ pour tout $t \ge 0$. En guise de discrétisation en espace du problème, on procède par différences finies avec un pas $\Delta x = 1/(N_x + 1)$. Le problème semi-discret consiste alors en un système d'EDO posé dans $\mathbb{R}^{N_x}$:

$$U'(t) = -AU(t) + S, \tag{5}$$

où l'on note $U(t) = (u_i(t))_{1 \le i \le N_x}$ tel que $u_i(t)$ approche $u(t, i\Delta x)$, et $u_0(t) = u_{N_x+1}(t) = 0$. Le second membre S vérifie $S_i = s(i\,\Delta x)$. La matrice A symétrique définie et positive déjà rencontrée maintes fois est

$$A = \frac{1}{\Delta x^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & 2 & -1 & 0 \\ \vdots & & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{pmatrix}.$$

1. Pour la donnée initiale $u(0,x) = \sin(\pi x) + \frac{1}{2}\sin(5\pi x)$ et le terme source $s(x) = 100\sin(5\pi x)$, résoudre le problème sur l'intervalle de temps $[0, 1]$ par la méthode d'Euler explicite. On utilisera $N_x = 10$ et successivement $h = 0.01$ puis $h = 0.001$. Tester ensuite des valeurs plus grandes de N_x.
2. Résoudre ce même problème avec la méthode d'**Euler implicite** :
$$\forall i \in \mathbb{N}, \quad y_{i+1} = y_i + h_i f(t_{i+1}, y_{i+1}).$$

Exercice 2. *Méthodes symplectiques*

Étant donné une fonction $H \in \mathcal{C}^1(\mathbb{R}^d \times \mathbb{R}^d; \mathbb{R})$, on définit le système hamiltonien

$$\begin{cases} p'(t) = -\nabla_q H(p,q), \\ q'(t) = \nabla_p H(p,q). \end{cases} \tag{6}$$

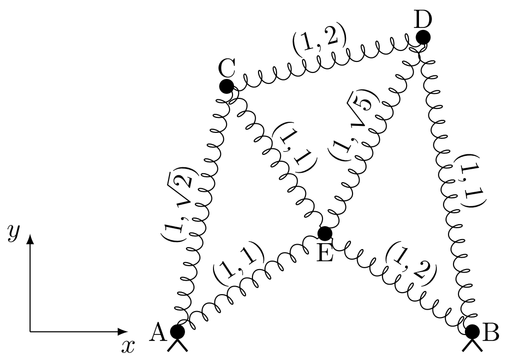
De nombreux systèmes de la physique prenne la forme précédente (cf. cours de Philippe Chartier). À titre d'exemple, citons le cas du mouvement d'une particule ponctuelle de masse m, de position $q \in \mathbb{R}^d$ et de moment $p \in \mathbb{R}^d$, soumise à une force F dérivant d'un potentiel $V : \mathbb{R}^d \longrightarrow \mathbb{R}$, avec donc $F = -\nabla V(q)$. Le principe fondamental de la dynamique s'écrit alors $mq'' = -\nabla V$, soit encore $mq' = p$ et $p' = -\nabla V(q)$. Le Hamiltonien concerné est alors l'énergie totale du système physique : $H(p,q) = \frac{1}{2m}|p|^2 + V(q)$ et le système différentiel prend la forme (6).

1. Vérifier brièvement que pour toute solution maximale $(p,q) \in \mathcal{C}^1(\mathbb{R}, \mathbb{R}^d \times \mathbb{R}^d)$, le hamiltonien $H(p,q)$ est constant au cours du temps. En déduire que si les ensembles de niveaux de H sont compacts, alors les trajectoires sont toutes globales.
2. On se place dans le cas de l'oscillateur harmonique : $H(p,q) = p^2 + q^2$ (ici $d = 1$). Comment l'énergie du système évolue-t-elle numériquement avec les méthodes d'Euler explicite et d'Euler implicite ?
3. Que se passe-t-il avec la méthode d'**Euler symplectique** suivante ?

$$\begin{cases} p_{i+1} = p_i - h_i \nabla_q H(p_i, q_{i+1}), \\ q_{i+1} = q_i + h_i \nabla_p H(p_i, q_{i+1}). \end{cases}$$

Application : le problème des ressorts

Un système de cinq masses assimilées à des points matériels $P \in \{A, B, C, D, E\}$ tous de même masse $m = 1$ sont reliés par sept ressorts comme l'indique le schéma suivant. Les points sont soumis aux seules forces issues de l'effet de ces ressorts, en particulier la gravité et les frottements éventuels ne sont pas considérés.



Chaque ressort r est caractérisé par une constante de raideur k_r et une longueur d'équilibre L_r^0 ; sa longueur effective dans la configuration courant est notée L_r. Les valeurs indiquées sur le schéma correspondent aux données (k_r, L_r^0). Les points de base $A = (0,0)$ et $B = (3,0)$ sont fixés dans le plan $\mathbb{R}^2$. L'énergie potentielle et l'énergie cinétique du système sont respectivement :

$$E_{\text{pot}} = \sum_r \frac{1}{2} k_r (L_r^0 - L_r)^2, \qquad E_{\text{kin}} = \sum_P \frac{1}{2} m \left| \frac{dP}{dt} \right|^2 .$$

Le système est hamiltonien avec :

$$\begin{aligned} q &= {}^t(x_C, y_C, x_D, y_D, x_E, y_E), \\ p &= {}^t(\dot{x}_C, \dot{y}_C, \dot{x}_D, \dot{y}_D, \dot{x}_E, \dot{y}_E), \\ H(p,q) &= E_{\text{pot}} + E_{\text{kin}}. \end{aligned}$$

1. Dans les fichiers `RessortLibrary.py` et `Example.py` disponibles en ligne à l'adresse ci-dessous, on trouvera des fonctions pré-programmées qui pourront être librement utilisées.

   `http://perso.eleves.ens-rennes.fr/people/pierre.le-barbenchon/TPanum.`
2. Déterminer par une méthode d'Euler explicite, la configuration du système à l'instant $t_f = 20$ avec un pas $h = 0.1$, pour la donnée de Cauchy $p^0 = 0_{\mathbb{R}^6}$ et $q^0 = {}^t(0.5, 1.5, 3.5, 1.5, 1, -0.5)$.
3. Tracer l'évolution de l'énergie potentielle, de l'énergie cinétique et de l'énergie totale du système au cours du temps, pour la solution numérique précédemment obtenue.
4. Reprendre ces deux questions en utilisant cette fois-ci une méthode d'Euler symplectique.
5. On ajoute une force de frottement de la forme $-\alpha p$ dans la seconde équation, avec $\alpha > 0$. Que donnent les deux méthodes précédentes (configuration géométrique et énergie totale) ? On pourra résoudre sur $[0, 100]$ avec un pas $h = 0.1$ et pour un coefficient de frottement $\alpha = 0.1$.

Fonctionnalité : Pour représenter la solution, on pourra faire une animation en fonction du temps en adaptant le code du fichier `Example.py` qui est en ligne.